PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04L 9/08 — A1

(11) International Publication Number: **WO 99/57844**

(43) International Publication Date: 11 November 1999 (11.11.99)

(21) International Application Number: PCT/CA99/00356

(22) International Filing Date: 3 May 1999 (03.05.99)

(30) Priority Data:

09/070,794	1 May 1998 (01.05.98)	US
2,236,495	1 May 1998 (01.05.98)	CA

(71) Applicant *(for all designated States except US)*: CERTICOM CORP. [CA/CA]; Suite 103, 200 Matheson Boulevard West, Mississauga, Ontario L5R 3L7 (CA).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: BLAKE-WILSON, Simon [GB/CA]; 237 Montrose Avenue, Toronto, Ontario M6G 3G6 (CA). JOHNSON, Donald, B. [US/US]; 4253 Sleepy Lake Drive, Fairfax, VA 22033 (US). MENEZES, Alfred [CA/CA]; Apartment 1604, 6 Willow Street, Waterloo, Ontario N2J 4S3 (CA).

(74) Agents: PILLAY, Kevin et al.; Orange Chari Pillay, Suite 3600, Toronto Dominion Bank Tower, Toronto-Dominion Centre, P.O. Box 190, Toronto, Ontario M5K 1H6 (CA).

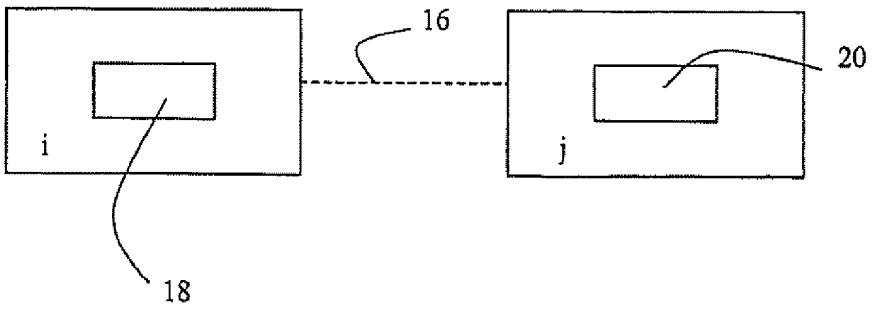
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published

With international search report.

Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: AUTHENTICATED KEY AGREEMENT PROTOCOL



(57) Abstract

A key agreement method between a pair of entities i and j in a digital data communication system, wherein each entity has a private and corresponding public key pair S_i,P_i and S_j,P_j respectively and the system, having global parameters for generating elements of a group, the method comprising the steps of: (a) entity i selecting a random private session value R_i; (b) forwarding a public session value corresponding to the private session value R_i to the entity j; (c) entity j computing a long term shared secret key k' derived from entity i's public key and j's private key utilizing a first function H_1; (d) the entity j utilizing entity j utilizing the key k' and computing an authenticated message on entity identities i, j and entities public session keys and forwarding the authenticated message to entity i; (e) the entity i verifying the received authenticated message; (f) the entity i computing the long term shared secret key k' derived from the entity j's public key and i's private key in accordance with the first function H_1; (g) the entity i utilizing the long term shared secret key k' and computing an authenticated message on the entities i and j identity information and the entities public session keys and forwarding the authenticated message to the entity j; (h) entity j verifying the received authenticated message; and (i) upon both the entities i and j verifying the authenticated message, computing a short term shared secret key utilizing a respective entity's session public and private keys.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

AUTHENTICATED KEY AGREEMENT PROTOCOL

This invention relates to cryptographic systems and in particular, to authenticated key agreement protocols used in the cryptographic systems.

A key agreement problem exists when two entities wish to agree on keying information in secret over a distributed network. Solutions to the key agreement problem whose security is based on a Diffie-Hellman problem in finite groups have been used extensively.

Suppose however, that entity *i* wishes to agree on secret keying information with entity *j*. Each party desires an assurance that no party, other than *i* and *j*, can possibly compute the keying information agreed upon. This may be termed the authenticated key agreement (AK) problem. Clearly, this problem is harder than the key agreement problem in which *i* does not care which entity it is agreeing on a key with, for in this problem *i* stipulates that the key may be shared with *j* and no other entity.

Several techniques related to the Diffie-Hellman problem have been proposed to solve the AK problem. However, no practical solutions have been provably demonstrated to achieve this goal and this deficiency has led, in many cases, to the use of flawed protocols.

Since in the AK problem, *i* merely desires that only *j* can possibly compute the key and not that *j* has actually computed the key, solutions are often said to provide implicit (key) authentication. If *i* wants to make sure, in addition, that *j* really has computed the agreed key, then key confirmation is incorporated into the key agreement protocol leading to so-called explicit authentication. The resulting goal is called authenticated key agreement with key confirmation (AKC). It may be seen that key confirmation essentially adds the assurance that *i* really is communicating with *j*. Thus, the goal of key confirmation is similar to the goal of entity authentication as defined in Diffie-Hellman. More precisely however, the incorporation of entity authentication into the AKA protocol provides i the additional assurance that *j* can compute the key, rather than the stronger assurance that *j* has actually computed the key.

A number of distinct types of attacks have been proposed against previous schemes. There are two major attacks which a protocol should withstand. The first is a passive attack, where an adversary attempts to prevent a protocol from achieving its goal by merely observing honest entities carrying out the protocol. The second is an active attack where an adversary additionally subverts the communication themselves in any way possible by injecting messages, intercepting messages, replaying messages, altering messages and the like.

It is thus essential for any secure protocol to withstand both passive and active attacks since an adversary can reasonably be assumed to have these capabilities in a distributed network.

It is therefore desirable to provide a key agreement protocol that mitigates at least some of the above advantages.

SUMMARY OF THE INVENTION

A key agreement method between a pair of entities i and j in a digital data communication system, wherein each said entity has a private and corresponding public key pairs S_i, P_i and S_j,P_j respectively and the system, having global parameters for generating elements of a group, said method comprising the steps of:

(a) entity *i* selecting a random private session value R_i;

(b) forwarding a public session value corresponding to said private session value R_i to said entity *j*;

(c) entity *j* computing a long term shared secret key *k'* derived from entity *i*'s public key and *j*'s private key utilizing a first function H_1;

(d) said entity *j* utilizing entity *j* utilizing said key *k'* and computing an authenticated message on entity identities *i*, *j* and entities public session keys and forwarding said authenticated message to entity *i*;

(e) said entity *i* verifying said received authenticated message;

(f) said entity *i* computing said long term shared secret key *k'* derived from said entity *j*'s public key and *i*'s private key in accordance with said first function H_1;

(g) said entity i utilizing said long term shared secret key *k'* and computing an authenticated message on said entities *i* and *j* identity information and said entities public session keys and forwarding said authenticated message to said entity *j*;

(h) entity *j* verifying said received authenticated message; and

(i) upon both said entities *i* and *j* verifying said authenticated message, computing a short term shared secret key utilizing a respective entity's session public and private keys.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features of the preferred embodiments of the invention will become more apparent in the following detailed description in which reference is made to the appended drawings wherein:

Figure 1 is a schematic diagram of a digital data communication system;

Figure 2, 3, 4 and 5, are embodiments of a key agreement protocol according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In the following discussion, the notation as outlined below is utilized and described more fully in the Diffie-Hellman paper entitled "New Directions in Cryptography", IEEE Transactions on Information Theory, November 1976, and incorporated herein by reference.

Referring to figure 1, a data communication system 10 includes a pair of entities or correspondents, designated as a sender *i* and a recipient *j* who are connected by a communication channel 16. Each of the correspondents, *i* and j includes an encryption unit that may process digital information and prepare it for transmission through channel 16 as will be described below. Furthermore, the encryption units may be either a dedicated processor or a general purpose processor including software for programming the general purpose computer to perform specific cryptographic functions.

In the following discussion, $k_{i,j}$ is *i*'s key pair k_i together with *j*'s public value, tran is a transcript of the ordered set of messages transmitted and received by *i* and *j* is the agreed key.

The protocols are described in terms of arithmetic operations in a subgroup generated by an element α of prime order q in the multiplicative group $Z_p^* = \{1,2,...,p-1\}$ where p is a prime. In each case, an entity's private value is an element $S_i of Z_q^* = \{1,2,..q-1\}$, and the corresponding public value is $P_i = \alpha^{S_i} MOD_p$, so that *i*'s key pair is $K_i = (S_i, P_i)$. It is to be noted that the protocols can be described equally well in terms of the arithmetic operations in any finite group and, of course, this would require the conversion of the security assumptions on the Diffie-Hellman problem to that group. Furthermore, any particular run of a protocol is called a session. For example, the keying information agreed in the course of a protocol run is referred to as a session key. The individual messages that form a protocol run are called flows.

The protocols as described below employ various primitives. Of these primitives, the two primitives used are message authentication codes (MAC) and Diffie-Hellman schemes

(DHS). Of course, some applications may wish to use another primitive to achieve confirmation. For example, if the agreed session key is later to be used for encryption, it seems sensible to employ an encryption scheme to achieve key confirmation rather than waste time implementing a MAC.

Referring now to figure 2, a graphical representation of a first embodiment of an AKC protocol (Protocol 1) according to the present invention is shown generally by the numeral 22. We use $\in_R$ to denote an element chosen independently at random, and commas to denote a unique encoding through concatenation (or any other unique encoding). Let H_1 and H_2 represent independent random oracles and (p,q,α) are global parameters. The random oracles may be defined in terms of coin tosses in the following way. It is assumed that all parties are provided with a black-box random function $H(\cdot): \{0,1\}^* \rightarrow \{0,1^k\}$. When H is queried for the first time, say on string x, it returns the string of length k corresponding to its first k coin tosses as H(x). When queried with a second string say x', first H compares x and x'. If $x' = n$, H again returns its first k coin tosses H(n). Otherwise H returns its second k tosses as H(x'). In instantiations, H is generally modeled by a hash function H. When entity i wishes to initiate a run of P with the entity j, i selects an element at random $R_i \in_R Z_q^*$ and sends α^{R_i} to j. On receipt of this string, j checks that $2 \le \alpha^{R_i} \le p-1$ and $\left(\alpha^{R_i}\right)^q = 1$, then j chooses $R_j \in_R Z_q^*$, and computes α^{Rj} and $k' = H_1\left(\alpha^{s_i s_j}\right)$. Finally, j uses k' to compute $MAC_{k'}\left(2, i, j, \alpha^{Rj}, \alpha^{Ri}\right)$, and sends this authenticated message to i. (Recall that $MAC_k(m)$ represents the pair (m,a), not just the tag a). On receipt of this string, i checks that the form of this message is correct, and that $2 \le \alpha^{R_j} \le p-1$ and $\left(\alpha^{Rj}\right)^q = 1$. The entity i then computes $k' = H_1(\alpha^{s_i s_j})$, recall that α^{s_j} is j's long term public key, and verifies the authenticated message it received. If so, i accepts, and sends back to j $MAC_{k'}\left(3, i, j, \alpha^{R_i}, \alpha^{R_j}\right)$. Upon receipt of this string, j checks the form of the message, verifies the authenticated message, and accepts. Both parties compute the agreed session key as $k = H_2\left(\alpha^{R_i R_j}\right)$. If at any stage, a check or verification performed by i or j fails, then that party terminates the protocol run, and rejects.

In practice, entity i may wish to append its identity to the first flow of Protocol 1. We omit this identity because certain applications may desire to identify the entities involved at the packet level rather than the message level - in this instance, identifying i again is therefore superfluous.

Note that entities use two distinct keys in Protocol 1 - one key for confirmation and a different key as the session key for subsequent use. In particular, the common practice of using the same key for both confirmation and as the session key may be disadvantageous if this means the same key is used by more than one primitive.

Protocol 1 is different from most proposed AKC protocols in the manner that entities employ their long-term secret values and session-specific secret values. Most proposed protocols use both long-term secrets and short-term secrets in the formation of all keys. In Protocol 1, long-term secrets and short-term secrets are used in quite independent ways. Long-term secrets are used only to form a session-independent confirmation key and short-term secrets only to form the agreed session key. Conceptually, this approach has both advantages and disadvantages over more traditional techniques. On the plus side, the use of long-term keys and short-term keys is distinct, serving to clarify the effects of a key compromise - compromise of a long-term secret is fatal to the security of future sessions, and must be remedied immediately, whereas compromise of a short-term secret effects only that particular session. On the negative side, both entities must maintain a long-term shared secret key k' in Protocol 1.

Protocol 2.

Protocol 2 is an AKC protocol designed to deal with some of the disadvantages of Protocol 1. It is represented graphically in figure 3. The actions performed by entities i and j are similar to those of Protocol 1, except that the entities use both their short-term and long-term values in the computation of both the keys they employ. Specifically, the entities use $k' = H_2\left(\alpha^{R_i R_j}, \alpha^{S_i S_j}\right)$ as their MAC key for this session, and $k = H_2\left(\alpha^{R_i R_j}, \alpha^{S_i S_j}\right)$ as the agreed session key. Unlike Protocol 1, both long-term secrets and both short-term secrets are used in Protocol 2 to form each key. While this makes the effect of a compromise of one of these values less clear, it also means that there is no long-term shared key used to MAC messages in every session between i and j. However, the two entities do still share a long-term secret value $\alpha^{S_i S_j}$. This value must therefore be carefully guarded against compromise, along with S_i and S_j themselves. Conceptually, it is possible to separate the AK phase and the key confirmation phase in Protocol 2.

Protocol 3. An embodiment of a secure AK protocol is illustrated in figure 4 which shows a graphical representation of the actions taken by i and j in a run of Protocol 3. To see that Protocol 3 is not a secure AK protocol if an adversary can reveal unconfirmed session keys, notice the following attack. E begins two runs of the protocol, one with $\Pi^3_{i,j}$, and one

with $\Pi^u_{i,j}$. Suppose $\Pi^3_{i,j}$ sends α^{R_i}, and $\Pi^u_{i,j}$ sends $\alpha^{R'_i}$. E now forwards α^{R_i} to $\Pi^u_{i,j}$, and $\alpha^{R'_i}$ to $\Pi^a_{i,j}$. E can now discover the session key $k = H_2\left(\alpha^{R_iR_j}, \alpha^{S_iS_j}\right)$ held by $\Pi^3_{i,j}$ by revealing the (same) key held by $\Pi^u_{i,j}$.

In this protocol, care must be taken when separating authenticated key agreement from key confirmation. Protocol 3 above is not a secure AK protocol in the full model of distributed computing, but can nonetheless be turned into a secure AKC protocol, as in Protocol 2. At issue here is whether it is realistic to expect that an adversary can learn keys that have not been confirmed.

Therefore, in this description we have tried to separate the goals of AK and AKC. A reason we have endeavored to separate authenticated key agreement from key confirmation is to allow flexibility in how a particular implementation chooses to achieve key confirmation. For example, architectural considerations may require key agreement and key confirmation to be separated - some systems may provide key confirmation during a 'real-time' telephone conversation subsequent to agreeing a session key over a computer network, while others may instead prefer to carry out confirmation implicitly by using the key to encrypt later communications.

The reason that we have specified the use of a subgroup of prime order by the DHSs is to avoid various known session key attacks on AK protocols that exploit the fact that a key may be forced to lie in a small subgroup of Z^*_p. From the point of view of the security proofs, we could equally well have made assumptions about DHSs defined in Z^*_p rather than a subgroup of Z^*_p.

It may be noted in particular that, as is the case with Protocol 3, many previous AK protocols do not contain asymmetry in the formation of the agreed key to distinguish which entity involved is the protocol's initiator, and which is the protocol's responder

Protocol 4. Again, in this protocol instead of describing the actions of *i* and *j* verbally, we illustrate these actions in figure 5. While at first glance, Protocol 4 may look almost identical to the well-known MTI protocol, where the shared value computed is $\alpha^{S_iR_j+S_jR_i}$, notice the following important distinction. Entity i calculates a different key in Protocol 4 depending on whether *i* believes it is the initiator or responder. In the first case, *i* computes $k = H_2\left(\alpha^{S_iR_j}, \alpha^{S_jR_i}\right)$, and in the second case $k = H_2\left(\alpha^{S_jR_j}, \alpha^{S_iR_j}\right)$. As we noted above, such asymmetry is desirable in a secure AK protocol. Of course, such asymmetry is not always

desirable -- a particular environment may require that *i* calculate the same key no matter whether *i* is the initiator or responder.

If indeed it can be shown that Protocol 4 is a secure AK protocol, then it can be turned into a secure AKC protocol in the same spirit as Protocol 2.

One issue is how to instantiate the random oracles H_1 and H_2. A hash function such as SHA-1 should provide sufficient security for most applications. It can be used in various ways to provide instantiations of independent random oracles. For example, an implementation of Protocol 1 may choose to use:

$$H_1(x) := SHA\text{-}1(01, x) \text{ and } H_2(x) := SHA\text{-}1(10, x) .$$

A particularly efficient instantiation of the random oracles used in Protocol 2 is possible using SHA-1 or RIPEMD-160. Suppose 80-bit session keys and MAC keys are required. Then the first 80 bits of $SHA\text{-}1(\alpha^{R_iR_j}, \alpha^{S_iS_j})$ can be used as k' and the second 80 bits used as k. Of course, such efficient implementations may not offer the highest conceivable security assurance of any instantiation.

It is easy to make bandwidth savings in implementations of the AKC protocols. Instead of sending the full authenticated messages (m,a) in flows 2 or 3, in both cases the entity can omit much of m, leaving the remainder of the message to be inferred by its recipient.

In some applications, it may not be desirable to carry out a protocol run each time a new session key is desired. Considering specifically Protocol 2 by way of example, entities may wish to compute the agreed key as:

$$H_2(\alpha^{R_iR_j}, \alpha^{S_iS_j}, counter) .$$

Then, instead of running the whole protocol each time a new key is desired, most of the time the counter is simply incremented. Entities need then only to resort to using the protocol itself every now and then to gain some extra confidence in the 'freshness' of the session keys they're using.

In Protocols 1, 2, and 3, performance and security reasons may make it desirable to use a larger (and presumably more secure) group for the static. Diffie-Hellman number $(\alpha_1^{S_iS_j})$ than for the ephemeral Diffie-Hellman number $(\alpha_2^{R_iR_j})$ calculation. The larger group is desirable because the static number will be used more often. The static numbers may be cached to provide a speed up in session key calculation.

Finally, note that a practical instantiation of G (assume G generates key pairs for each entity) using certificates should check knowledge of the secret value before issuing a

certificate on the corresponding public value. We believe that this is a sensible precaution in any implementation of a Certification Hierarchy.

While the invention has been described in connection with the specific embodiment thereof, and in a specific use, various modifications thereof will occur to those skilled in the art without departing from the spirit of the invention as set forth in the appended claims. For example, each entity will usually generate key pairs itself and then get them certified by a certification authority.

The terms and expressions which have been employed in this specification are used as terms of description and not of limitations, there is no intention in the use of such terms and expressions to exclude any equivalence of the features shown and described or portions thereof, but it is recognized that various modifications are possible within the scope of the claims to the invention.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A key agreement method between a pair of entities i and j in a digital data communication system, wherein each said entity has a private and corresponding public key pairs S_i, P_i and S_j,P_j respectively and the system, having global parameters for generating elements of a group, said method comprising the steps of:

(a) entity *i* selecting a random private session value R_i;

(b) forwarding a public session value corresponding to said private session value R_i to said entity *j*;

(c) entity *j* computing a long term shared secret key *k'* derived from entity *i*'s public key and *j*'s private key utilizing a first function H_1;

(d) said entity *j* utilizing entity *j* utilizing said key *k'* and computing an authenticated message on entity identities *i, j* and entities public session keys and forwarding said authenticated message to entity *i*;

(e) said entity *i* verifying said received authenticated message;

(f) said entity *i* computing said long term shared secret key *k'* derived from said entity *j*'s public key and *i*'s private key in accordance with said first function H_1;

(g) said entity i utilizing said long term shared secret key *k'* and computing an authenticated message on said entities *i* and *j* identity information and said entities public session keys and forwarding said authenticated message to said entity *j*;

(h) entity *j* verifying said received authenticated message; and

upon both said entities *i* and *j* verifying said authenticated message, computing a short term shared secret key utilizing a respective entity's session public and private keys.

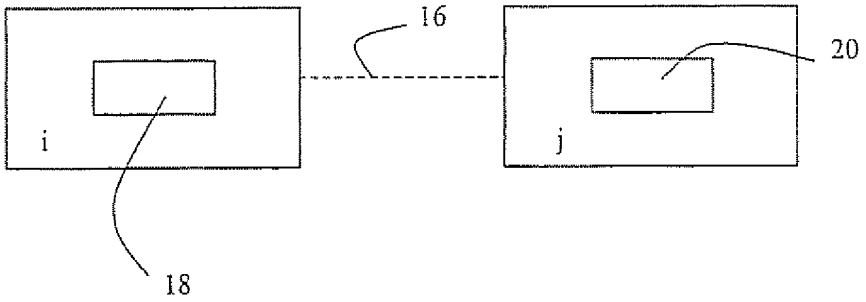


Figure 1

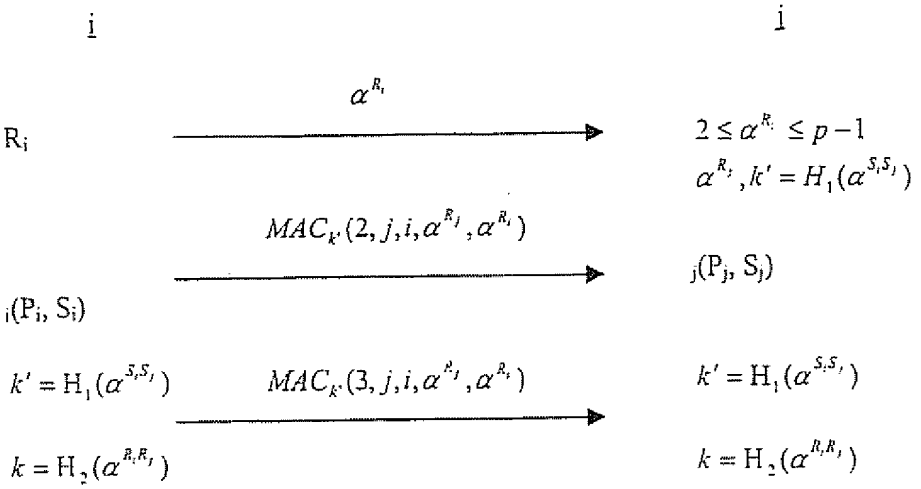


Figure 2 Protocol 1

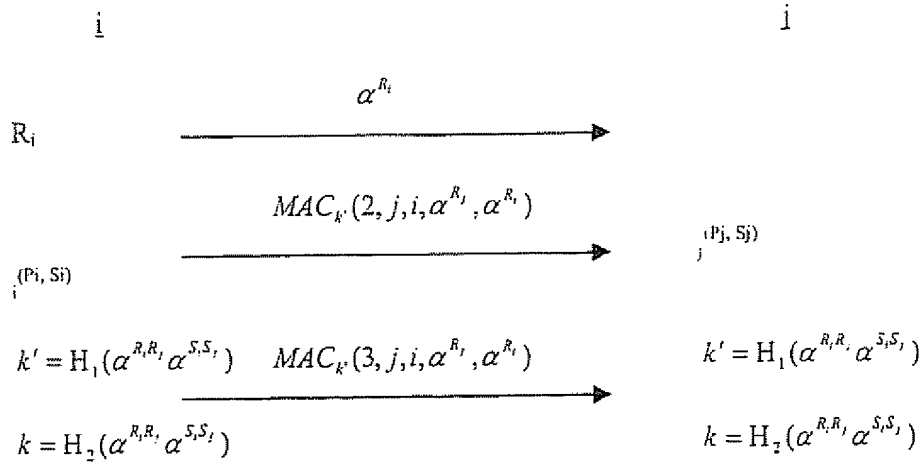


Figure 3 Protocol 2

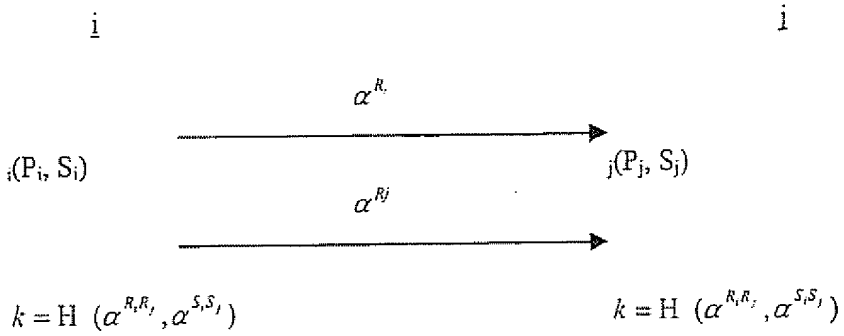


Figure 4 Protocol 3

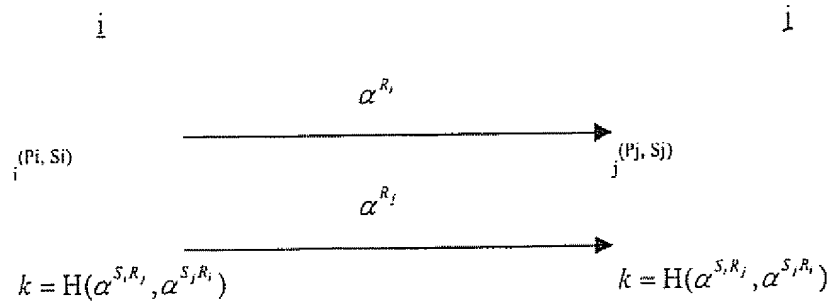


Figure 5 Protocol 4

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/CA 99/00356

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04L9/08

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 481 121 A (SIEMENS AG) 22 April 1992 (1992-04-22) abstract page 2, line 11 - line 20 page 3, line 3 - page 4, line 31 figures 2,3 claim 1 ---	1
A	JABLON D P: "STRONG PASSWORD-ONLY AUTHENTICATED KEY EXCHANGE" COMPUTER COMMUNICATIONS REVIEW, vol. 26, no. 5, 1 October 1996 (1996-10-01), pages 5-26, XP000641968 ISSN: 0146-4833 page 8, line 4 - page 10, line 11 page 19, line 1 - page 20, last last -----	1

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
1 September 1999	10/09/1999

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Gautier, L

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Int tional Application No
PCT/CA 99/00356

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0481121 A	22-04-1992	NONE	